SharpZO: Hybrid Sharpness-Aware Vision Language Model Prompt Tuning via Forward-Only Passes

Yifan Yang[1], Zhen Zhang[1], Rupak Vignesh Swaminathan[2], Jing Liu[2], Nathan Susanj[2], Zheng Zhang[1]
[1]University of California, Santa Barbara
[2]Amazon AGI
{yifanyang, zhen_zhang}@ucsb.edu {swarupak, jlmk, nsusanj}@amazon.com
zhengzhang@ece.ucsb.edu

Abstract

Fine-tuning vision language models (VLMs) has achieved remarkable performance across various downstream tasks; yet, it requires access to model gradients through backpropagation (BP), making them unsuitable for memory-constrained, inference-only edge devices. To address this limitation, previous work has explored various BP-free fine-tuning methods. However, these approaches often rely on high-variance evolutionary strategies (ES) or zeroth-order (ZO) optimization, and often fail to achieve satisfactory performance. In this paper, we propose a hybrid Sharpness-aware Zeroth-order optimization (SharpZO) approach, specifically designed to enhance the performance of ZO VLM fine-tuning via a sharpness-aware warm-up training. SharpZO features a two-stage optimization process: a sharpness-aware ES stage that globally explores and smooths the loss landscape to construct a strong initialization, followed by a fine-grained local search via sparse ZO optimization. The entire optimization relies solely on forward passes. Detailed theoretical analysis and extensive experiments on CLIP models demonstrate that SharpZO significantly improves accuracy and convergence speed, achieving up to 7% average gain over state-of-the-art forward-only methods. [1]

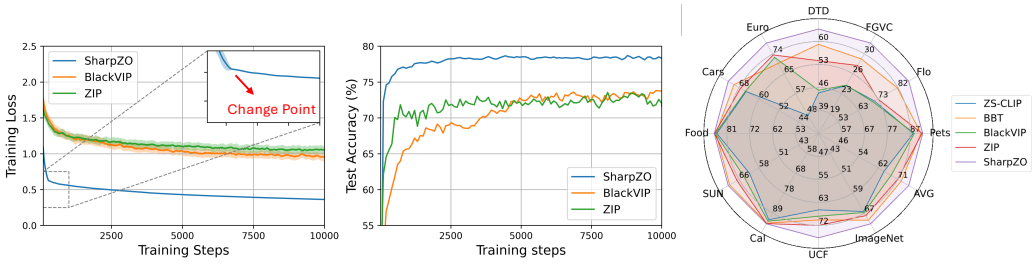
1 Introduction

In recent years, fine-tuning vision-language models (VLMs) has achieved remarkable performance across a wide range of downstream tasks, including image classification [53, 54], object detection [11, 51], and image segmentation [44, 23]. Among these models, one of the most prominent is CLIP [33], which has attracted significant attention for its powerful zero-shot recognition capabilities. To further improve the performance of VLMs in downstream tasks, previous work has explored the use of efficient, trainable prompt parameters [54, 48, 42] for the prompt tuning of VLMs. However, these prompt-tuning techniques are heavily dependent on the availability of a backward computation engine, which is typically unavailable on memory-constrained edge devices used in Internet-of-Things (IoT) applications [39] or wearable technologies [8].

To address these limitations, recent studies have explored fine-tuning VLMs in backpropagation-free settings [37, 48, 42]. These approaches optimize trainable prompts by leveraging high-variance black-box optimization techniques such as Evolutionary Strategies (ES) [17, 2] and Zeroth-Order (ZO) optimization [30, 31, 52] as alternatives to the first-order (FO) methods used in white-box scenarios. For instance, [48] employ ES to update prompt parameters by evaluating sampled prompts through forward passes only, thereby eliminating the need for memory-expensive back propagation.

[1]Code available at `https://github.com/yifanycc/sharpzo`.

Preprint. Under review.



(a) Curve for Training Loss and Testing Accuracy on EuroSAT (b) Downstream Generalization

Figure 1: a) Comparison between SharpZO and other ZO prompt-tuning baselines.SharpZO demonstrates significantly lower variance than other ZO-based baselines like ZIP [31] and BlackVIP [30]. (b) Fine-tuned performance across all 11 tasks tested compared with ZIP and BlackVIP and BBT [38]. All experiments are conducted using the CLIP model with a ViT-B/16 backbone.

More recently, ZO stochastic gradient descent (SGD) [14] methods have been adapted to VLM fine-tuning in the work of BlackVIP and ZIP [30, 31]. By approximating gradients with just two forward evaluations, these ZO approaches avoid the high computational cost and instability of ES, yet still match its performance while requiring substantially fewer model queries [31].

However, existing ZO-based VLM fine-tuning methods remain substantially inferior to backpropagation-based training. Their high variance and inherently local search dynamics make them prone to premature convergence in suboptimal regions. Previous work has attempted to improve the performance of ZO optimization by reducing the problem dimensionality through pruning [15, 25, 49] and low-rank decomposition [45, 31] of the trainable parameters. However, in widely adopted prompt-tuning settings, such parameter reduction offers limited benefit, as the number of trainable parameters is already inherently small in original trainable prompt.

In contrast to prior work that reduces the variance of ZO optimization by limiting the number of trainable parameters, our approach introduces a new perspective that focuses on initialization and the sharpness of the loss landscape. Specifically, we propose SharpZO, a hybrid Sharpness-Aware Zeroth-Order optimization method that employs a two-stage framework to significantly reduce the variance of ZO gradient estimation and improve the performance of ZO-based VLMs prompt tuning.

The first stage performs warm-up training using a sharpness-aware Covariance Matrix Adaptation Evolution Strategy (CMA-ES), which provides both a smoother loss landscape and a strong initialization for the second stage. Unlike gradient-based methods that follow local descent directions, CMA-ES enables effective global exploration by adaptively shaping the search distribution based on past evaluations [28]. Moreover, incorporating sharpness not only improves model generalization but also improves the accuracy of the randomized gradient estimators used in stage 2 ZO training, which is unbiased only with respect to a smoothed version of the objective function [14].

In the second stage, we perform fine-grained local optimization using a sparse Zeroth-Order Stochastic Gradient Descent (ZO-SGD) method. To further reduce gradient estimation variance, we introduce a novel *Z-pruning* technique specifically designed for noisy ZO settings, effectively reducing the dimensionality of the search space. Unlike conventional magnitude-based pruning used in previous sparse ZO method [15, 25], Z-pruning leverages gradient information to capture the influence of model non-linearity and applies Z-score-based normalization [12] to suppress outlier gradient estimates.

As shown in Figure 1, our method converges faster with significantly lower variance compared with other ZO prompt-tuning baselines, achieving up to a 7% average improvement in accuracy. Our main contributions are summarized as follows:

- We propose SharpZO, a novel hybrid sharpness-aware optimizer that fine-tunes VLMs using only forward passes. To our knowledge, this is the first ZO method that improves performance considering the sharpness-aware initialization.
- In the first stage, we introduce a sharpness-aware CMA-ES that enhances generalization and reduces second stage ZO gradient estimation variance by smoothing the loss landscape.
- In the second stage, we develop a sparse ZO fine-tuning method with a novel Z-pruning technique to suppress outliers in noisy gradient estimates.
- We validate SharpZO through extensive experiments and theoretical analysis, demonstrating superior performance over existing BP-free baselines.

2 Background

2.1 Coordinate-wise Gradient Estimation or Randomized Gradient Estimation?

Mainstream ZO gradient estimation methods can be broadly classified into two categories: Coordinate-wise Gradient Estimation (CGE) and Randomized Gradient Estimation (RGE). In our SharpZO framework, we employ CGE to compute sharpness-related terms in stage 1 and pruning metrics in stage 2, while RGE is used to update parameters during the second-stage ZO-SGD optimization. Below, we provide background on both approaches and highlight their key differences in terms of estimation variance and computational efficiency.

Given a VLM with trainable prompt vector $\boldsymbol{w} \in \mathbb{R}^d$, we define the training cross-entropy loss as $\mathcal{L}(\boldsymbol{w})$. ZO estimated gradients $\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{w})$ are estimated via forward differences between function evaluations, where the perturbation of the trainable parameters $\boldsymbol{w}$ depends on whether the CGE or RGE method is used, which gives:

$$\textbf{(RGE)}\ \hat{\nabla}\mathcal{L}(\boldsymbol{w}) = \frac{1}{q}\sum_{i=1}^{q}\left[\frac{\mathcal{L}(\boldsymbol{w}+\mu\boldsymbol{u}_i)-\mathcal{L}(\boldsymbol{w}-\mu\boldsymbol{u}_i)}{2\mu}\boldsymbol{u}_i\right];\ \textbf{(CGE)}\ \hat{\nabla}\mathcal{L}(\boldsymbol{w}) = \sum_{i=1}^{d}\left[\frac{\mathcal{L}(\boldsymbol{w}+\mu\boldsymbol{e}_i)-\mathcal{L}(\boldsymbol{w}-\mu\boldsymbol{e}_i)}{2\mu}\boldsymbol{e}_i\right]. \quad (1)$$

Here, $\mu > 0$ is a smooth parameter, $\boldsymbol{u}_i \in \mathbb{R}^d$ denotes a randomized perturbation vector perturbing all parameters at the same time in RGE, and $e_i = [0, 0, \cdots, 1, \cdots, 0]^T$ represents the i-th standard basis vector, which is used to compute a finite-difference approximation of $\mathcal{L}(\boldsymbol{w})$ along a single coordinate in CGE. To reduce the variance of the RGE gradients, it is common to average the gradients estimated over q different randomized perturbations, where q is called query numbers. In contrast, the CGE method approximates the gradient by perturbing individual coordinates and estimating the directional derivative along each axis using finite differences.

Unlike FO methods that compute exact gradients $\nabla\mathcal{L}(\boldsymbol{w})$ via BP, RGE methods estimate gradients in a biased manner toward the exact gradients, which instead provides an unbiased estimate of the gradient of a smoothed version of the objective, defined as $\mathcal{L}_\mu(\boldsymbol{w}) = \mathbb{E}_{\boldsymbol{u}}[\mathcal{L}(\boldsymbol{w}+\mu\boldsymbol{u})]$. In contrast, CGE estimates directional derivatives along individual coordinates without applying such smoothing, resulting in greater sensitivity to sharp changes in the loss landscape.

Difference between RGE and CGE: We compare RGE and CGE primarily in terms of query complexity and accuracy. The number of function queries differs significantly between the two methods. Given the dimension of trainable parameter as d and the number of RGE query as q, RGE requires $\mathcal{O}(q)$ queries, whereas CGE incurs a higher cost of $\mathcal{O}(d)$ queries. Clearly, RGE offers much lower query complexity, especially when $q = 1 \ll d$ in our case. Despite its higher query complexity, CGE achieves superior accuracy compared to RGE, as it directly approximates the true gradient without introducing smoothed objective function [5].

2.2 Sharpness-aware Optimization

Sharpness-Aware Minimization (SAM) [13] was originally proposed to improve generalization by smoothing the loss landscape and encouraging the optimization process to converge to flat minima. The key idea is to minimize the worst-case loss within a neighborhood around the current parameters $\boldsymbol{w}$ by introducing an analytic approximation of the worst-case perturbation $\boldsymbol{\epsilon}^*$ within a radius ρ. Specifically, the SAM objective is formulated as:

$$\min_{\boldsymbol{w}} \mathcal{L}_\rho^{\text{SAM}}(\boldsymbol{w}), \quad \text{where} \quad \mathcal{L}_\rho^{\text{SAM}}(\boldsymbol{w}) = \max_{\|\boldsymbol{\epsilon}\|_2 \le \rho} \mathcal{L}(\boldsymbol{w}+\boldsymbol{\epsilon}).$$

Following the objective function, the gradient approximation for SAM after dropping the second-order terms is given as $\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{w}')|_{\boldsymbol{w}'+\boldsymbol{\epsilon}^*}$ by computing the worst-case perturbation $\boldsymbol{\epsilon}^*$. To estimate $\boldsymbol{\epsilon}^*$, SAM approximates the inner maximization using a FO Taylor expansion of the loss function. This yields the following analytic solution:

$$\boldsymbol{\epsilon}^* \approx \arg\max_{\|\boldsymbol{\epsilon}\|_2 \le \rho} \big(\mathcal{L}(\boldsymbol{w}) + \boldsymbol{\epsilon}^\top \nabla\mathcal{L}(\boldsymbol{w})\big) = \rho \cdot \nabla\mathcal{L}(\boldsymbol{w})/\|\nabla\mathcal{L}(\boldsymbol{w})\|_2. \quad (2)$$

Unlike previous works that apply SAM to replace standard gradient descent either throughout the entire training process [13] or during the final few epochs [55], we, for the first time, investigate the effectiveness of incorporating sharpness information into CMA-ES as an early-stage warm-up strategy to enhance the performance of ZO fine-tuning.

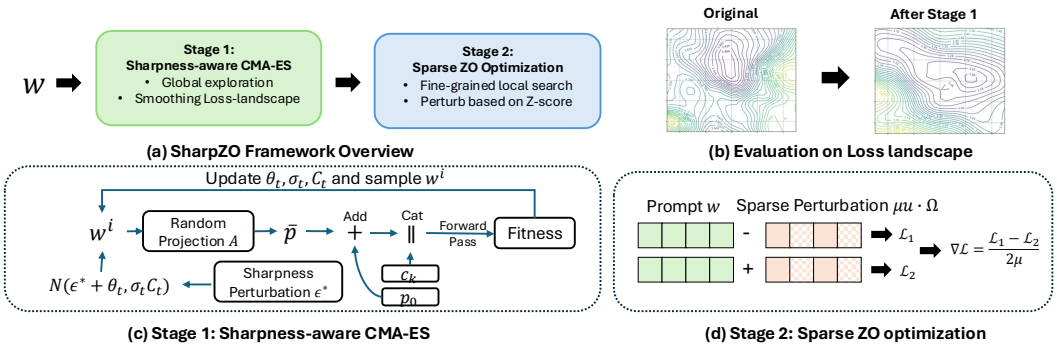


Figure 2: Overview of the SharpZO method. (a) The overall training pipeline of SharpZO, consisting of a two-stage optimization process. (b) Visualization of the smoothed loss landscape after Stage 1 sharpness-aware CMA-ES optimization. (c) Training dynamics of the sharpness-aware CMA-ES method. (d) RGE-based gradient estimation during sparse ZO training in Stage 2.

3 The SharpZO Method

In this section, we detail the SharpZO method, which is designed to fine-tune VLMs using only forward passes. As illustrated in Fig. 2 (a), our approach consists of two main stages: a sharpness-aware CMA-ES stage and a sparse ZO fine-grained search stage. We demonstrate that performing a sharpness-aware global search in the early steps of training significantly enhances the performance of ZO optimization—both in terms of convergence speed and final accuracy. In the following, we first summarize the problem setup of this work and then describe each stage of our method in detail.

We consider a black-box VLM with loss function $\mathcal{L}(\boldsymbol{w})$ and a dataset $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^{N}$ with a total of N samples and K classes. For models like CLIP, it is necessary to construct a text prompt for each class. Specifically, for a given class $k \in \{1, \ldots, K\}$ and the model hidden dimension m, the class-specific prompt $\boldsymbol{p}^k$ is defined by concatenating a predefined initial text embedding $\boldsymbol{p}_0 \in \mathbb{R}^m$ (e.g., "a photo of a") with the class label embedding $\boldsymbol{c}^k$, yielding $\boldsymbol{p}^k = [\boldsymbol{p}_0, \boldsymbol{c}^k]$. To make the prompt $\boldsymbol{p}^k$ trainable, we introduce a parameter $\bar{\boldsymbol{p}}_t$ that modifies the initial embedding $\boldsymbol{p}_0$, where $t \in [1, T]$ is the training steps. This parameter is obtained via a random projection from a low-dimensional trainable matrix $\boldsymbol{w}_t \in \mathbb{R}^d$, where $d \ll m$ is the latent dimension. Then, a fixed randomized projection matrix $\boldsymbol{A} \in \mathbb{R}^{m \times d}$ is used to project $\boldsymbol{w}$ into the embedding space, producing $\bar{\boldsymbol{w}}_t = \boldsymbol{w}_t \boldsymbol{A}^\top \in \mathbb{R}^m$ to matching the shape of the original initialized prompt $\boldsymbol{p}_0$. Thus, the overall training objective becomes:

$$\min_{\boldsymbol{w}} \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}([\boldsymbol{p}_0 + \bar{\boldsymbol{p}}_t, c_k], (x_n, y_n)), \quad \text{where} \quad \bar{\boldsymbol{p}} = \boldsymbol{w}_t \boldsymbol{A}^\top$$

Next, we provide a detailed introduction to the two stages of the SharpZO method separately.

3.1 Stage 1: Sharpness-Aware CMA-ES Method

In this section, we first summarize the traditional CMA-ES method and then propose our sharpness-aware CMA-ES optimization for a warm-up training.

In the traditional CMA-ES method, given the population size S, the optimizer generates a population of candidate solutions $\boldsymbol{w}_t^i, i \in [1, S]$, where $\boldsymbol{w}_t^i$ is obtained by sampling from the current multivariate normal distribution $\boldsymbol{w}_t^i \sim \boldsymbol{\theta}_t + \sigma_t \mathcal{N}(0, \boldsymbol{C}_t)$. Here, $\boldsymbol{\theta}_t$ is the mean of the distribution, σ is the step size, S is the population size and C is the covariance matrix capturing the shape and scale of the search distribution. After evaluating the loss (fitness) $\mathcal{L}(\boldsymbol{w}_t^i, \mathcal{D})$ of these samples by forwarding the training samples $\mathcal{D}$ along with the trainable prompt $\boldsymbol{w}_t^i, i \in [1, S]$, the parameters $\boldsymbol{\theta}_t$, σ_t, and $\boldsymbol{C}_t$ are updated accordingly [17].

Different from the previous CMA-ES method, we propose a new sharpness-aware CMA-ES method to smooth the loss landscape during the stage 1 warm-up training, which help to reduce the stage 2 gradient estimation accuracy. Specifically, we add the worst-case perturbation $\boldsymbol{\epsilon}^*$ during the sampling

of CMA-ES method, where $\boldsymbol{\epsilon}^*$ is computed within a local Euclidean ball based on eq. (2) and gives:

$$\boldsymbol{w}^i \sim \boldsymbol{\epsilon}^* + \boldsymbol{\theta}_t + \delta_t \mathcal{N}(0, \boldsymbol{C}_t), \quad i \in [1, S], \quad \boldsymbol{\epsilon}^* = \rho \frac{\nabla \mathcal{L}(\boldsymbol{w})}{\|\nabla \mathcal{L}(\boldsymbol{w})\|_2}. \tag{3}$$

The effectiveness of this modification can be explained through the Taylor expansion of the Monte Carlo estimation for the loss $\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_t + \boldsymbol{\epsilon}^* + o)]$ used in the sharpness-aware CMA-ES optimizer, given $o \sim \mathcal{N}(0, \delta_t^2 \boldsymbol{C}_t)$. Given the sampling strategy in eq. (3) and omitting the first-order term by the fact $\mathbb{E}[z] = 0$, the expected fitness can be approximated as:

$$\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_t + \boldsymbol{\epsilon}^* + o)] \approx \mathcal{L}(\boldsymbol{\theta}_t + \boldsymbol{\epsilon}^*) + \frac{1}{2}\mathbb{E}[o^\top \nabla^2 \mathcal{L}(\boldsymbol{\theta}_t + \boldsymbol{\epsilon}^*) o], \quad o \sim \mathcal{N}(0, \delta_t^2 \boldsymbol{C}_t).$$

As observed, in addition to optimizing the same term $\mathcal{L}(\boldsymbol{\theta}_t + \boldsymbol{\epsilon}^*)$ as gradient-based SAM methods, the effectiveness of the sharpness-aware CMA-ES approach is achieved with additional higher order term involving stochastic adaptation of the covariance matrix, which introduces an additional mechanism to explore and down-weight high-curvature directions.

Another challenge in applying sharpness-aware CMA-ES lies in gradient estimation. Specifically, we cannot directly access the gradient information needed to compute $\boldsymbol{\epsilon}^*$ due to the absence of backpropagation. Unlike prior work [50, 47], which relies on RGE-based gradient estimation with a large query budget q, we adopt CGE for this purpose. CGE provides an unbiased estimate of the gradient along each coordinate, making it more suitable for computing the sharpness-aware perturbation term, as discussed in Section 2.1. The detailed formulation for estimating $\nabla \mathcal{L}(\boldsymbol{w})$ using CGE is provided in eq. (1).

3.2 Stage 2: Fine-grained ZO Method

After obtaining a strong initialization and a smoothed loss landscape through the first-stage global search, we proceed to optimize the parameters toward the global optimum using a ZO-SGD method. In this stage, we perform fine-grained and efficient local search guided by RGE-estimated gradients. To further reduce the variance, we propose a new Z-pruning metrics with pruning mask Ω to reduce the effective dimension during gradient estimation.

Unlike prior sparse ZO methods that rely solely on magnitude-based pruning metrics, we introduce a novel pruning criterion tailored to the high-variance nature of ZO-estimated gradients. In contrast to first-order sparse training approaches, which apply the pruning mask Ω directly to model weights, we apply the mask to the perturbation vector $\boldsymbol{u}$. This strategy effectively reduces the dimensionality during ZO perturbation and results in the following gradient estimator:

$$\hat{\nabla}_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}) = \frac{\mathcal{L}(\boldsymbol{w} + \mu \cdot \Omega \boldsymbol{u}) - \mathcal{L}(\boldsymbol{w} - \mu \cdot \Omega \boldsymbol{u})}{2\mu} \boldsymbol{u} \tag{4}$$

We note that a query number of $q = 1$, as defined in Eq. (1), is sufficient for stable training of our SharpZO method when preceded by the Stage 1 warm-up. This is significantly more efficient than prior works such as ZIP [31] and BLACKVIP [30], which still exhibit high training variance even with a query number of $q = 5$, as shown in Fig. 1. Next, we introduce how we construct the Z-pruning based pruning mask Ω.

Z-pruning Metrics: The Z-pruning metric is designed to minimize the loss degradation introduced by pruning, by considering the sensitivity of each parameter. Specifically, considering $\delta \boldsymbol{w}$ reflect the change of weights during pruning, the difference in loss between the dense and pruned models can be approximated via a first-order Taylor expansion:

$$\mathcal{L}(\boldsymbol{w} + \delta \boldsymbol{w}) - \mathcal{L}(\boldsymbol{w}) = \nabla \mathcal{L}(\boldsymbol{w})^\top \delta \boldsymbol{w} + \frac{1}{2} \delta \boldsymbol{w}^\top H \delta \boldsymbol{w} + \mathcal{O}(\|\delta \boldsymbol{w}\|^3),$$

where H denotes the Hessian of the loss with respect to the parameters. Here, we can approximate the Hessian H using a Fisher matrix as $H \approx \mathbb{E}_{x \sim \mathcal{D}}[\nabla \mathcal{L}(\boldsymbol{w}; x)^2]$ and estimating gradients $\nabla \mathcal{L}(\boldsymbol{w}; x)$ using the CGE method described in eq. (1) as $\hat{\nabla} \mathcal{L}(\boldsymbol{w}; x)$. By considering the second-order term, we can obtain the Z-pruning metrics as:

$$\Omega = |\boldsymbol{w}|^2 \cdot z(\mathbb{E}_{x \sim \mathcal{D}}[\hat{\nabla} \mathcal{L}(\boldsymbol{w}, x)^2]), \quad \text{where } \hat{\nabla} \mathcal{L}(\boldsymbol{w}) = \sum_{i=1}^{d} \left[\frac{\mathcal{L}(\boldsymbol{w} + \mu \boldsymbol{e}_i) - \mathcal{L}(\boldsymbol{w} - \mu \boldsymbol{e}_i)}{2\mu} \boldsymbol{e}_i \right], \tag{5}$$

Algorithm 1 SharpZO: Hybrid Sharpness-Aware Zeroth-Order Optimization

Require: Initial prompt parameters $\boldsymbol{w}_0$, total steps T, transition step T_c, pruning interval K

for $t = 1$ to T **do**
  if $t < T_c$ **then** ▷ Stage 1: Sharpness-aware CMA-ES
    Sample candidate solutions $\boldsymbol{w}_t^i$ using eq. (3)
    Evaluate fitness $\mathcal{L}(\boldsymbol{w}_t^i)$ for each candidate
    Update CMA-ES parameters $\boldsymbol{\theta}_t$, σ_t, and $\boldsymbol{C}_t$ based on fitness values
  else ▷ Stage 2: Sparse ZO Optimization
    if $t = T_c$ or $(t - T_c) \bmod K = 0$ **then**
      Update pruning mask Ω using eq. (5)
    end if
    Estimate gradient $\hat{\nabla}\mathcal{L}(\boldsymbol{w}_t)$ using the ZO oracle (eq. 4)
    $\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t - \eta \cdot \hat{\nabla}\mathcal{L}(\boldsymbol{w}_t)$
  end if
end for
return Fine-tuned prompt vector $\boldsymbol{w}_T$

where $z(\boldsymbol{g}^2) = (\boldsymbol{g}^2 - \mu_g)/\sigma_g$ denote the Z-score normalization given μ_g and σ_g as the mean and standard deviation of the gradient vector $\boldsymbol{g} = \hat{\nabla}\mathcal{L}$. The Z-score standardizes the gradient magnitudes to mitigate the scale mismatch between trainable parameters and their gradients—a mismatch that is especially pronounced in ZO settings. We consider adapting the second-order term as the pruning score based on the practice of previous pruning work [36, 46]. In Section 5.4.2, we demonstrate the effectiveness of the Z-pruning method compared to dense and magnitude-based pruned ZO training.

3.3 Algorithms

We present the SharpZO algorithm in Algorithm 1. In practice, we first execute Stage 1 for a total of T_c steps, followed by Stage 2 until the total training budget of T steps is reached. The transition point T_c is determined automatically using a strategy inspired by early stopping, based on the observed change in validation accuracy. Typically, $T_c \ll T$ and is typically reached within 100 steps. As a result, although Stage 1 is more computationally expensive per step, the overall training efficiency remains high. During Stage 2, we update the pruning mask every K steps to balance computational cost and adaptability to the evolving optimization landscape. The learning rate for ZO optimization in Stage 2 is denoted by η.

4 Theoretical Guarantee

In this section, we give a quick proof to show the convergence rate of SharpZO method. By comparing the SharpZO convergence rate with the baseline ZO-SGD rate given in MeZO [27], we highlight why our hybird smoothness-aware setup can significantly help to improve the performance of VLMs fine-tuning. To align our analysis with VLMs/LLMs fine-tuning, we consider a non-convex optimization setup and the proof assume the loss landscape follows the Polyak-Lojasiewicz (PL) inequality, which has been widely considered in other ZO fine-tuning papers [27]. First, we list the following assumptions for our analysis include the PL-inequality we just mentioned:

A1 (PL Inequality): The loss function ℓ satisfies the Polyak–Łojasiewicz (PL) condition. That is, there exists a constant $\mu > 0$ such that for all $\boldsymbol{w} \in \mathbb{R}^d$, we have $\frac{1}{2}\|\nabla\mathcal{L}(\boldsymbol{w})\|^2 \geq \mu\,(\mathcal{L}(\boldsymbol{w}) - \mathcal{L}^*)$, where $\mathcal{L}^*$ denotes the global minimum of the loss function.

A2 (Lipschitz smoothness): The loss function $\mathcal{L}$ has an L-Lipschitz continuous gradient. That is, there exists a constant $L > 0$ such that for all $\boldsymbol{w}_i, \boldsymbol{w}_j \in \mathbb{R}^d$, we have $\|\nabla\mathcal{L}(\boldsymbol{w}_i) - \nabla\mathcal{L}(\boldsymbol{w}_j)\| \leq L\|\boldsymbol{w}_i - \boldsymbol{w}_j\|$.

Theorem 1. *Under assumptions A1 and A2, suppose the SharpZO algorithm first performs T_c steps of global optimization using CMA-ES and then switches to zeroth-order gradient-based optimization until convergence. The convergence rate of SharpZO method can be give by:*

$$t \approx T_c + \mathcal{O}\left(\frac{1}{\eta\mu}\log\left(\frac{L^3\eta^2\rho^2}{\epsilon}\right)\right) \tag{6}$$

Table 1: Few-shot performance across 11 datasets using CLIP models with ResNet and ViT backbones, trained for 20K steps. * indicate results reported in prior works [30, 42, 31]. We additionally reproduce the ZIP results, as the original paper restricted the query budget to 5K. Bold values highlight the best performance, demonstrating the superiority of SharpZO over all BP-free baselines.

Backbone	Methods	ImageNet	Pets	Flo	FGVC	DTD	Euro	Cars	Food	SUN	Cal	UCF	AVG
RN50	ZS-CLIP*	58.18	85.77	66.14	17.28	42.32	37.56	55.61	77.31	58.52	86.29	61.46	58.77
	BBT*	61.74	88.73	72.53	12.07	54.33	69.01	60.24	78.44	64.34	90.05	67.91	65.40
	BlackVIP	60.33	85.99	65.12	17.37	42.73	58.16	56.70	77.23	59.17	86.37	60.11	60.84
	ZIP	61.30	**89.53**	68.41	19.98	47.40	63.10	58.61	78.98	62.86	90.63	64.05	64.08
	SharpZO	**63.29**	89.51	**79.50**	**23.97**	**60.58**	**80.77**	**60.58**	**79.28**	**66.17**	**91.24**	**72.43**	**69.76**
ViT-B/16	ZS-CLIP*	66.73	89.21	71.34	24.72	44.39	47.60	65.32	86.06	62.50	92.94	66.75	65.23
	BBT*	70.15	92.70	82.41	29.49	59.26	70.48	70.19	86.42	70.33	94.75	70.48	72.42
	BlackVIP*	67.10	89.70	70.60	24.78	45.20	73.10	65.60	86.60	64.70	93.70	69.10	68.20
	ZIP (Offical)*	66.20	94.00	**92.30**	26.80	47.80	64.60	71.09	86.40	63.30	94.00	69.80	70.57
	ZIP (Rep)	68.35	93.18	73.00	28.32	54.26	74.19	67.58	87.01	67.43	94.97	72.51	70.98
	SharpZO	**71.60**	**94.06**	88.02	**32.34**	**63.95**	**79.42**	**72.50**	**87.13**	**70.86**	**95.09**	**77.08**	**75.64**

where ϵ is given by assuming $\mathcal{L}(\boldsymbol{w}_t) - \mathcal{L}^* \leq \epsilon$, η is the learning rate of ZO-SGD optimizer in stage 2 and L is the smoothness factor. Eq. (6) is obtained by ignoring the lower order terms for clarity.

Proof. Details of the proof can be found in Appendix D. □

Compared to the naive ZO-SGD convergence rate presented in [27], the SharpZO method leverages a sharpness-aware initialization strategy that yields a lower starting point for the second-stage ZO training, specifically of the form $(1 - \mu(1 - 2L\sigma^2))^{T_c}\,\Delta_0$ in eq. (13) in Appendix D. Since we use a relative large step size σ during the first stage training, we can observe the original error gap Δ_0 is linearly decreasing with high scaling factor. Moreover, as we can observe from the sharpness term in eq. (6), the integration of sharpness-aware optimization effectively clips the L-smoothness constant L with the sharpness parameter ρ, thereby reducing the effective sharpness in the ZO training phase.

5 Experiments

In this section, we present experimental results to evaluate the performance of the proposed SharpZO method across a variety of downstream tasks using CLIP models with different architectures. Specifically, we compare the proposed SharpZO method with zero-shot (ZS) inference and other BP-free baselines like BBT [48], BlackVIP [30], and ZIP [31] (Detailed descriptions for tasks and baslines method can be found in Appendix B). Our results demonstrate that SharpZO not only achieves superior accuracy but also improves efficiency, as measured by the time-to-test-accuracy (ToTA) metric [7]. Additionally, we provide a comprehensive ablation study to analyze the contributions of individual components in Section 5.4.1 and Section 5.4.2. Further implementation details and extended experimental results—including evaluations across various model architectures and hyper-parameter choices, as well as comparisons with state-of-the-art prompt-tuning methods that involve backpropagation, such as CraFT [42] are provided in Appendix C.

Training Detail: For the VLM model, we utilize CLIP [33] with both ResNet [18] and ViT [10] backbones as the visual encoder, and Transformers [40] as the text encoder. The CLIP weights are initialized from the official pretrained checkpoints and remain frozen during training. The prompt generator use initial prompt with length of 4, and hidden dimension $d = 512$. Parameters in $\boldsymbol{w}$ are initialized from a Gaussian distribution $\mathcal{N}(0, 0.02)$. To determine the transition point T_c between the two training stages, we adopt an early stopping criterion: the algorithm switches to stage 2 if the validation accuracy does not improve by more than 0.01 over the best recorded accuracy for 10 consecutive steps. Detail hyper-parameter setup for SharpZO method on various tasks can be found in Table. 7 in Appendix C.2.

5.1 Results on Few-Shot Classification

We first compare our SharpZO method with SOTA BP-free prompt-tuning baselines across 11 downstream tasks. To explore the effect of different model architectures, we evaluate all methods using CLIP models with both ResNet-50 and ViT-B/16 viusal encoder backbones. The results are summarized in Table 1. Based on these results, we draw the following conclusions:

Table 2: Comparison of robustness to distribution shift between SharpZO and other baselines. The best results are highlighted in bold.

Method	ResNet-50					ViT-B/16				
	ImageNet	-V2	-Sketch	-A	-R	ImageNet	-V2	-Sketch	-A	-R
ZS-CLIP	58.2	51.3	33.3	21.7	56.0	66.7	60.8	46.2	47.8	74.0
BBT	61.7	54.0	33.9	23.2	58.3	70.2	63.0	47.9	49.5	76.1
BlackVIP	60.2	52.3	33.3	21.5	57.7	65.5	59.2	44.6	42.5	73.1
ZIP	61.3	53.7	33.7	23.9	57.6	68.4	59.7	45.5	47.1	75.2
SharpZO	**63.3**	**54.8**	**35.2**	**24.5**	**58.7**	**71.6**	**63.8**	**45.0**	**50.3**	**76.6**

SharpZO significantly outperforms all other BP-free methods. As shown in Table 1, SharpZO consistently surpasses other ZO prompt tuning approaches in terms of classification accuracy. Compared to the SOTA ZO prompt tuning method ZIP, SharpZO achieves an absolute average performance gain of **5%** and outperforms ZIP among **all** 11 tasks on the CLIP model with ViT-B/16 backbone. These improvements are driven by the reduction of gradient estimation variance and bias with the sharpness-aware warm-up training.

SharpZO performs robustly across diverse model architectures. Unlike prior ZO prompt-tuning methods such as ZIP and BlackVIP—which often struggle to converge on certain tasks like Flowers102, EuroSAT, and UCF101 when using CLIP models with ResNet backbones—our proposed SharpZO method consistently delivers strong performance across a wide range of architectures and tasks. Additional results using architectures such as ResNet-101 and ViT-B/32, presented in Appendix C.1, further demonstrate the robustness of SharpZO to varying model backbones.

SharpZO exhibits lower training variance and improved generalization. As shown in the training curves in Fig. 1 (a), SharpZO demonstrates significantly lower variance during the optimization process compared to ZIP and BlackVIP. Moreover, compared to methods that rely on naive CMA-ES optimization, SharpZO achieves substantially better generalization, largely due to the incorporation of sharpness-aware information in the warm-up stage.

5.2 Robustness to Distribution Shift

In this section, we further evaluate the robustness of the SharpZO method under distribution shifts. Results comparing SharpZO to other BP-free baselines are summarized in Table 2. Compared to the state-of-the-art ZO method ZIP, SharpZO achieves an absolute improvement of **2.0%** on ResNet-50 and **2.8%** on ViT-B/16 (averaged over all distribution shift benchmarks) for the ImageNet test accuracy. These findings highlight the strong out-of-distribution generalization ability of SharpZO under varying types of distribution shifts.

5.3 Time-to-test-accuracy Efficiency

In this section, we evaluate the training efficiency of the proposed SharpZO method. We focus on training time rather than memory usage, as all zeroth-order (ZO) baselines exhibit comparable memory consumption due to their forward-only nature. Specifically, we measure the wall-clock time required to reach a common evaluation accuracy threshold, following the protocol of [7]. The threshold is selected such that it is attainable by all baselines. The results are summarized in Table 3 and tested on single Nvidia A100-40G GPU.

Methods	**IN**	**Pets**	**DTD**	**Euro**
BlackVIP	172.6	714.8	132.0	201.5
ZIP	19.0	126.3	6.0	251.4
SharpZO	**15.3**	**2.4**	**2.6**	**12.7**

Table 3: Time-to-test accuracy comparison between different BP-free prompt tuning methods on multiple dataset. The time is recorded in minutes.

As shown in Table 3, the SharpZO method achieves faster convergence compared to other ZO prompt tuning baselines, which is consistent with our theoretical analysis in Section 4. Beyond the benefits of the proposed hybrid sharpness-aware optimization scheme, the improved training speed of SharpZO also stems from its lower per-step query count and significantly faster forward pass.

Specifically, unlike ZIP and BlackVIP, which require 10 queries per step to reduce training loss, SharpZO only requires 2 queries per step during Stage 2. Moreover, ZIP incurs substantial overhead due to its complex reconstruction process in forward pass, taking approximately 0.53 seconds per

Figure 3: Comparison between the naive CMA-ES with the Sharpness-aware (S-aware) CMA-ES method on EuroSAT dataset.

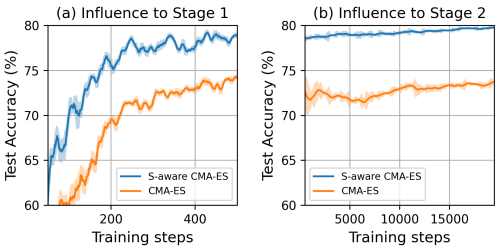
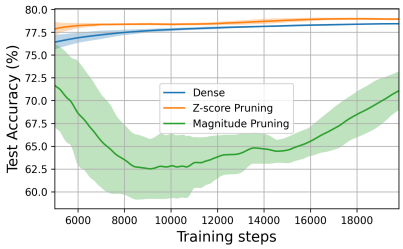


Figure 4: Comparison between the naive ZO optimization and sparse ZO optimization with various pruning metrics on EuroSAT dataset.



forward pass, whereas SharpZO requires only 0.0069 seconds. Consequently, the average per-step training time of SharpZO is markedly lower than other ZO baseline like ZIP.

5.4 Ablation on Components in Different Stages

5.4.1 Influence of Stage 1 Sharpness Aware Optimization

In this section, we aim to validate the effectiveness and illustrate the influence of our sharpness-aware CMA-ES method to both stage 1 and stage 2 training in SharpZO. Specifically, we compare the training curve between the naive CMA-ES method and our sharpness-aware (S-aware) CMA-ES method for both stage in Fig. 3 (a) and Fig. 3 (b), respectively. The experiments are conducted on the EuroSAT dataset using the CLIP model with a ViT-B/16 backbone. For the convenience of comparison, the transition point from Stage 1 to Stage 2 optimization is fixed at 500 steps.

As illustrated in Fig. 3 (a), the sharpness-aware CMA-ES method consistently achieves a faster convergence rate and superior final accuracy compared to the naive CMA-ES method in stage 1, which shows a better generalization ability. More importantly, the sharpness-aware training benefits the second-stage convergence of ZO optimization as observed from Fig. 3. The sharpness-aware warm-up training leads to a more stable Stage 2 training curve with reduced variance, which can be attributed to the implicitly clipped smoothness factor introduced by the sharpness-aware updates, as discussed in the theoretical analysis in Section 4.

5.4.2 Effectiveness of Stage 2 Sparse ZO Optimization

We evaluate the effectiveness of our proposed Z-pruning strategy for sparse ZO optimization in Stage 2. Specifically, we compare it against magnitude-based pruning and dense training on the EuroSAT dataset using the CLIP ViT-B/16 backbone, with consistent learning rate settings. Results are shown in Fig. 4.

Our findings show that sparse ZO training with Z-pruning reduces gradient variance and improves both accuracy and convergence speed compared to dense training. In contrast, magnitude-based pruning—commonly used in prior work [15, 25]—performs poorly in prompt-tuning due to the limited number of trainable parameters (512), which makes accurate pruning critical. Moreover, magnitude-based pruning operates solely on weight values, ignoring critical nonlinear interactions within the model. This limitation is particularly impactful in prompt-based tuning, where the prompts are prepended to the input and play a disproportionately large role in the model's behavior compared to standard weights.

6 Conclusion

This paper has introduced SharpZO, a hybrid ZO fine-tuning method comprising two optimization stages. In Stage 1, SharpZO employs a sharpness-aware CMA-ES algorithm to conduct a global search for optimal regions while simultaneously smoothing the loss landscape. In Stage 2, SharpZO performs fine-grained sparse ZO optimization for local optimization. Compared with prior BP-free fine-tuning approaches, SharpZO provides a high-performance, inference-only fine-tuning solution tailored for VLMs. Future work may explore the extension of SharpZO to full-model fine-tuning for both text- and multimodal LLMs.

References

[1] Youhei Akimoto, Yuichi Nagata, Isao Ono, and Shigenobu Kobayashi. Bidirectional relation between cma evolution strategies and natural evolution strategies. In *Parallel Problem Solving from Nature, PPSN XI: 11th International Conference, Kraków, Poland, September 11-15, 2010, Proceedings, Part I 11*, pages 154–163. Springer, 2010.

[2] Anne Auger and Nikolaus Hansen. Tutorial cma-es: evolution strategies and covariance matrix adaptation. In *Proceedings of the 14th annual conference companion on Genetic and evolutionary computation*, pages 827–848, 2012.

[3] Dara Bahri, Hossein Mobahi, and Yi Tay. Sharpness-aware minimization improves language model generalization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7360–7371, 2022.

[4] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part VI 13*, pages 446–461. Springer, 2014.

[5] Aochuan Chen, Yimeng Zhang, Jinghan Jia, James Diffenderfer, Jiancheng Liu, Konstantinos Parasyris, Yihua Zhang, Zheng Zhang, Bhavya Kailkhura, and Sijia Liu. Deepzero: Scaling up zeroth-order optimization for deep model training. *arXiv preprint arXiv:2310.02025*, 2023.

[6] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014.

[7] Cody Coleman, Daniel Kang, Deepak Narayanan, Luigi Nardi, Tian Zhao, Jian Zhang, Peter Bailis, Kunle Olukotun, Chris Ré, and Matei Zaharia. Analysis of dawnbench, a time-to-accuracy machine learning performance benchmark. *ACM SIGOPS Operating Systems Review*, 53(1):14–25, 2019.

[8] Erika Covi, Elisa Donati, Xiangpeng Liang, David Kappel, Hadi Heidari, Melika Payvand, and Wei Wang. Adaptive extreme edge computing for wearable devices. *Frontiers in Neuroscience*, 15:611300, 2021.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

[11] Yu Du, Fangyun Wei, Zihe Zhang, Miaojing Shi, Yue Gao, and Guoqi Li. Learning to prompt for open-vocabulary object detection with vision-language model. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14084–14093, 2022.

[12] Nanyi Fei, Yizhao Gao, Zhiwu Lu, and Tao Xiang. Z-score normalization, hubness, and few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 142–151, 2021.

[13] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021.

[14] Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM journal on optimization*, 23(4):2341–2368, 2013.

[15] Wentao Guo, Jikai Long, Yimeng Zeng, Zirui Liu, Xinyu Yang, Yide Ran, Jacob R Gardner, Osbert Bastani, Christopher De Sa, Xiaodong Yu, et al. Zeroth-order fine-tuning of llms with extreme sparsity. In *2nd Workshop on Advancing Neural Network Training: Computational Efficiency, Scalability, and Resource Optimization (WANT@ ICML 2024)*, 2024.

[16] Nikolaus Hansen and Anne Auger. Evolution strategies and cma-es (covariance matrix adaptation). In *Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation*, GECCO Comp '14, page 513–534, New York, NY, USA, 2014. Association for Computing Machinery.

[17] Nikolaus Hansen, Sibylle D Müller, and Petros Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary computation*, 11(1):1–18, 2003.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[19] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

[20] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8340–8349, 2021.

[21] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15262–15271, 2021.

[22] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.

[23] Zihan Li, Yunxiang Li, Qingde Li, Puyang Wang, Dazhou Guo, Le Lu, Dakai Jin, You Zhang, and Qingqi Hong. Lvit: language meets vision transformer in medical image segmentation. *IEEE transactions on medical imaging*, 43(1):96–107, 2023.

[24] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[25] Yong Liu, Zirui Zhu, Chaoyu Gong, Minhao Cheng, Cho-Jui Hsieh, and Yang You. Sparse mezo: Less parameters for better performance in zeroth-order llm fine-tuning. *arXiv preprint arXiv:2402.15751*, 2024.

[26] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[27] Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D Lee, Danqi Chen, and Sanjeev Arora. Fine-tuning language models with just forward passes. *Advances in Neural Information Processing Systems*, 36:53038–53075, 2023.

[28] Gregory Morse and Kenneth O Stanley. Simple evolutionary optimization can rival stochastic gradient descent in neural networks. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, pages 477–484, 2016.

[29] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008.

[30] Changdae Oh, Hyeji Hwang, Hee-young Lee, YongTaek Lim, Geunyoung Jung, Jiyoung Jung, Hosik Choi, and Kyungwoo Song. Blackvip: Black-box visual prompting for robust transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24224–24235, 2023.

[31] Seonghwan Park, Jaehyeon Jeong, Yongjun Kim, Jaeho Lee, and Namhoon Lee. ZIP: An efficient zeroth-order prompt tuning for black-box vision-language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[32] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012.

[33] Alec Radford, Jong Wook Kim, Christopher Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

[34] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pages 5389–5400. PMLR, 2019.

[35] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. A dataset of 101 human action classes from videos in the wild. *Center for Research in Computer Vision*, 2(11):1–7, 2012.

[36] Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. A simple and effective pruning approach for large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[37] Tianxiang Sun, Zhengfu He, Hong Qian, Yunhua Zhou, Xuan-Jing Huang, and Xipeng Qiu. Bbtv2: Towards a gradient-free future with large language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3916–3930, 2022.

[38] Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. Black-box tuning for language-model-as-a-service. In *International Conference on Machine Learning*, pages 20841–20855. PMLR, 2022.

[39] Nazli Tekin, Ahmet Aris, Abbas Acar, Selcuk Uluagac, and Vehbi Cagri Gungor. A review of on-device machine learning for iot: An energy perspective. *Ad Hoc Networks*, 153:103348, 2024.

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[41] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *Advances in neural information processing systems*, 32, 2019.

[42] Zhengbo Wang, Jian Liang, Ran He, Zilei Wang, and Tieniu Tan. Connecting the dots: Collaborative fine-tuning for black-box vision-language models. In *International Conference on Machine Learning*, pages 50931–50943. PMLR, 2024.

[43] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010.

[44] Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model. In *European Conference on Computer Vision*, pages 736–753. Springer, 2022.

[45] Yifan Yang, Kai Zhen, Ershad Banijamali, Athanasios Mouchtaris, and Zheng Zhang. Adazeta: Adaptive zeroth-order tensor-train adaption for memory-efficient large language models fine-tuning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 977–995, 2024.

[46] Yifan Yang, Kai Zhen, Bhavana Ganesh, Aram Galstyan, Goeric Huybrechts, Markus Müller, Jonas M. Kübler, Rupak Vignesh Swaminathan, Athanasios Mouchtaris, Sravan Babu Bodapati, Nathan Susanj, Zheng Zhang, Jack FitzGerald, and Abhishek Kumar. Wanda++: Pruning large language models via regional gradients. In *ICLR workshop in Sparsity in LLMs (SLLM)*, 2025.

[47] Feiyang Ye, Yueming Lyu, Xuehao Wang, Masashi Sugiyama, Yu Zhang, and Ivor Tsang. Sharpness-aware black-box optimization. In *The Thirteenth International Conference on Learning Representations*, 2025.

[48] Lang Yu, Qin Chen, Jiaju Lin, and Liang He. Black-box prompt tuning for vision-language model as a service. In *IJCAI*, pages 1686–1694, 2023.

[49] Zhen Zhang, Yifan Yang, Kai Zhen, Nathan Susanj, Athanasios Mouchtaris, Siegfried Kunzmann, and Zheng Zhang. Mazo: Masked zeroth-order optimization for multi-task fine-tuning of large language models. *arXiv preprint arXiv:2502.11513*, 2025.

[50] Huaqin Zhao, Jiaxi Li, Yi Pan, Shizhe Liang, Xiaofeng Yang, Wei Liu, Xiang Li, Fei Dou, Tianming Liu, and Jin Lu. Helene: Hessian layer-wise clipping and gradient annealing for accelerating fine-tuning llm with zeroth-order optimization. *arXiv preprint arXiv:2411.10696*, 2024.

[51] Shiyu Zhao, Zhixing Zhang, Samuel Schulter, Long Zhao, BG Vijay Kumar, Anastasis Stathopoulos, Manmohan Chandraker, and Dimitris N Metaxas. Exploiting unlabeled data with vision and language models for object detection. In *European conference on computer vision*, pages 159–175. Springer, 2022.

[52] Jiajun Zhou, Yifan Yang, Kai Zhen, Ziyue Liu, Yequan Zhao, Ershad Banijamali, Athanasios Mouchtaris, Ngai Wong, and Zheng Zhang. Quzo: Quantized zeroth-order fine-tuning for large language models. *arXiv preprint arXiv:2502.12346*, 2025.

[53] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16816–16825, 2022.

[54] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022.

[55] Zhanpeng Zhou, Mingze Wang, Yuchen Mao, Bingrui Li, and Junchi Yan. Sharpness-aware minimization efficiently selects flatter minima late in training. In *The Thirteenth International Conference on Learning Representations*, 2025.

A Limitations

While SharpZO demonstrates strong empirical and theoretical advantages for forward-only VLM fine-tuning, several limitations remain. First, the method is currently tailored for prompt-tuning scenarios with relatively low-dimensional parameter spaces; its scalability to full-model or multi-modal fine-tuning remains unexplored. Second, the sharpness-aware CMA-ES warm-up stage requires coordinate-wise gradient estimation (CGE), which may be computationally expensive for higher-dimensional settings. As a result, the SharpZO method proposed in this work is a better fit under the parameter-efficient fine-tuning setup.

B Further Detail Regarding Tasks and Baselines

Datasets: Following the experimental setup of prior VLMs fine-tuning works [42, 31], we evaluate SharpZO on 11 diverse image classification benchmarks under a few-shot learning scenario. These datasets cover a broad range of tasks: generic object recognition with ImageNet [9] and Caltech101 [41], fine-grained image classification with OxfordPets [32], StanfordCars [22], Flowers102 [29], Food101 [4], and FGVCAircraft [26], satellite image classification with EuroSAT [19], texture recognition with DTD [6], scene classification with SUN397 [43], and action recognition with UCF101 [35]. To assess the robustness of SharpZO under distribution shift, we further evaluate it on four widely-used out-of-distribution (OOD) variants of ImageNet: ImageNetV2 [34], ImageNet-Sketch [41], ImageNet-A [21], and ImageNet-R [20].

Baselines: To benchmark the performance of SharpZO against SOTA methods, we mianly consider five baseline approaches:

- **Zero-shot (ZS)**: This baseline uses manually crafted prompts to directly evaluate the pretrained CLIP model without any additional adaptation.
- **BBT** [48]: BBT employs a naive CMA-ES-based optimizer to update the trainable prompt parameters. As the original BBT is designed for LLMs, we adopt its prompt generator structure and adapt it to the VLM fine-tuning setting.
- **BlackVIP** [30]: BlackVIP uses a naive ZO-RGE estimator to jointly optimize both textual and visual prompts in a black-box manner.
- **ZIP** [31]: ZIP improves upon naive ZO prompt tuning by reducing the number of trainable parameters via low-rank decomposition of the prompt space.
- **CraFT** [42]: CraFT introduces a trainable adapter appended to the output of the CLIP model. It jointly optimizes both the prompt parameters and the adapter using a combination of CMA-ES and gradient-based methods. As CraFT requires access to backpropagation, we provide a separate comparison with it in Appendix C.3.

C Additional Experimental Results

C.1 Robustness across different model architectures

We further evaluate the performance of SharpZO across different model architectures and compare it with other baselines, with results summarized in Table 4. As shown, SharpZO demonstrates architecture-agnostic effectiveness, consistently outperforming previous backpropagation-free (BP-free) methods across all four evaluated architectures. In particular, SharpZO achieves an average absolute performance improvement of **2.25%** and **4.19%** over the ZIP and BlackVIP methods, respectively.

Table 4: Ablation study for model architectures with Imagenet dataset.

Methods	RN50	RN101	Vit-B/16	Vit-B/32	Avg.
ZS-CLIP	58.18	61.62	66.73	62.05	62.15
BlackVIP	60.33	62.00	67.10	61.10	62.63
ZIP	61.30	63.67	68.35	64.97	64.57
SharpZO	**63.29**	**65.40**	**71.60**	**66.98**	**66.82**

C.2 Hyper-parameter Search

To guide future applications of the SharpZO method, we conduct ablation studies on several key hyper-parameters, including the scaling factor for the sharpness term, and the sparsity ratio used in sparse ZO optimization. The results are summarized in Table 5.

Based on our experiments, the optimal scaling factor for the sharpness term should be around 0.05 or 0.1, which is consistent with the choice in the original SAM paper for the SAM-SGD algorithm. The sparsity ratio should be larger than 0.5. Conversely, a sparsity ratio that is too small will hurt the representational capacity of the prompt parameters, since the number of training parameters is already low enough during prompt tuning. This observation differs from previous ZO full-model fine-tuning work.

Table 5: Ablation studies for different applicable pruning metrics and sparsity.

Methods\Sparsity	10%	30%	50%	70%	90%
Magnitude	78.07	77.14	76.12	77.06	77.09
Z-Score	78.91	79.02	79.42	79.15	79.21

Table 6: Ablation studies for different scaling factor ρ.

ρ	0.001	0.01	0.05	0.1	0.5
Accuracy	77.74	79.03	79.22	79.42	76.32

Another hyperparameter that influences performance is the scaling factor ρ, which adjusts the weight of the sharpness-aware term during the sampling process in Stage 1 optimization. We conduct an ablation study with different values of ρ, as shown in Table 6.

We outline the configuration details for each comparative baselines. Specifically, the hyper-parameter setup of individual tasks for SharpZO method are presented in Table 7. For a fair comparison, we use the **original** hyperparameter settings provided in the baseline papers [31, 30, 42] when running the experiments. In contrast, for the ablation studies, we adopt a consistent parameter setup across methods to ensure comparability. Specifically, we would like to note that during the stage 2 of our method, we set the query number q as 1 instead of 5 used in previous baselines like ZIP and BlackVIP, which is enough for the convergence of SharpZO method.

Table 7: Hyper-parameter setup for the stage 1 and 2 for the SharpZO paper.

Method	Dataset	Pets	Flo	FGVC	DTD	Euro	Cars	Food	SUN	Cal	UCF	IN
SharpZO (Stage 1)	Step size σ	0.1	0.4	0.2	0.4	0.4	1.0	0.1	0.4	0.4	0.1	1.0
	ZO-CGE scale μ_{cge}	1e-3	1e-3	1e-3	1e-5	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3	1e-5
	Implicit population S						40					
	Intrinsic dimension d						512					
	Context tokens number m						4					
	Scaling factorρ						0.1					
	Population size S						40					
SharpZO (Stage 2)	Learning rate η	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3	1e-1	1e-3	1e-3	1e-1
	ZO-CGE scale μ_{cge}						1e-5					
	ZO-RGE scale μ_{rge}						1e-3					
	Pruning Interval K						200					
	Number of query q						1					
	Pruning ratio						0.5					

C.3 Comparison with Method Involving Backpropagation

In this section, we compare our method with black-box tuning baselines that involve backpropagation, such as CraFT [42]. The CraFT method introduces a collaborative fine-tuning framework that jointly optimizes both the prompt and the adapter, using ES for the former and first-order (FO) methods for the latter. Although CraFT achieves strong performance, its requirement of backpropagation limits its applicability in memory-constrained environments, such as mobile devices and edge devices, where gradient access via backpropagation is not available.

It is important to highlight that reliance on backpropagation presents significant challenges for deployment on edge inference devices, as these devices are typically equipped with inference-only ASICs and do not support gradient computation. Unlike large-scale multimodal models like LLaVA

[24], the CLIP model is particularly relevant for edge computing scenarios. Despite the inherent limitations of backpropagation-based methods in such contexts, we include CraFT in our comparison to demonstrate that our proposed SharpZO method can achieve even better performance without relying on backpropagation, and in a more efficient manner.

The comparison results on few-shot task accuracy and training efficiency, measured by time-to-accuracy (ToTA), are summarized in Table 8. As shown, SharpZO consistently outperforms CraFT across all evaluated tasks. Moreover, SharpZO also requires less training memory, as it avoids storing backpropagation graphs, further enhancing its suitability for edge deployment.

Table 8: Comparison between SharpZO and black-box fine-tuning baseline involving backpropagation. * represent accuracy results obtained from original CraFT paper [42]. We bold the best results during the compasion.

Methods	**Test Accuracy**				**Memory (MB)**			
	Imagenet	Pets	DTD	Euro	Imagenet	Pets	DTD	Euro
CraFT*	68.21	91.94	63.28	72.07	3297.5	3130.9	3128.7	2178.8
SharpZO	**71.60**	**93.46**	**63.95**	**79.42**	**3132.3**	**3032.5**	**3057.9**	**2075.2**

D Proofs

To prove Theorem 1, we begin by establishing Lemma 1, which characterizes the convergence of the first-stage sharpness-aware CMA-ES method by leveraging its interpretation as a natural-gradient descent algorithm [1]. We then apply the result of Lemma 1 as the initial condition for analyzing the convergence of the second-stage ZO optimization. Finally, by composing these two phases, we obtain the overall convergence rate of the SharpZO algorithm.

D.1 Proof of Lemma 1

Before presenting the Lemma 1, we first introduce some background knowledge regarding the connection between the CMA-ES update and the natural gradient descent, details about the mathematical relationship can be refereed to [1, 16].

Considering the minimization

$$\arg\min_{\boldsymbol{\theta}_t, \boldsymbol{C}_t} \; \mathbb{E}\big[\mathcal{L}(P) \,\big|\, \boldsymbol{\theta}_t, \boldsymbol{C}_t\big]$$

under the sampling distribution of eq. (3), a natural-gradient descent step with step-size σ reads

$$\theta_{t+1} = \boldsymbol{\theta}_t \; - \; \sigma\, \mathbf{F}_{\boldsymbol{\theta}_t}^{-1}\, \nabla_{\boldsymbol{\theta}_t} \mathbb{E}\big[\mathcal{L}(P) \,\big|\, \boldsymbol{\theta}_t, \boldsymbol{C}_t\big], \tag{7}$$

$$C_{t+1} = \boldsymbol{C}_t \; - \; \sigma\, \nabla_{\boldsymbol{C}_t} \mathbb{E}\big[\mathcal{L}(P) \,\big|\, \boldsymbol{\theta}_t, \boldsymbol{C}_t\big]. \tag{8}$$

where F_t^{-1} is the Fisher information matrix $\mathbf{F}_{\boldsymbol{\theta}_t} = -\mathbb{E}_{P\sim f(\cdot|\boldsymbol{\theta}_t)}\left[\frac{\partial^2 \log f(P|\boldsymbol{\theta}_t)}{\partial\theta\, \partial\theta^\top}\right]$. given the sample distribution defined in eq. (3). Here we have used that the natural gradient with respect to $\boldsymbol{\theta}_t$ satisfies: $\widetilde{\nabla}_{\boldsymbol{\theta}_t}\, \mathbb{E}[\mathcal{L}(P)] = \boldsymbol{C}_t^{-1}\, \nabla_{\boldsymbol{\theta}_t}\, \mathbb{E}[\mathcal{L}(P)]$.

To simplify the subsequent convergence proof, we note that our target is the expected fitness $f(m_t)$ of the sample center $m_t = \boldsymbol{\theta}_t$, whereas $\boldsymbol{C}_t$ affects only the sampling spread and not directly the objective value. In the idealized infinite-samples regime of CMA-ES one shows

$$\boldsymbol{C}_t^{-1} \;\propto\; \nabla^2 \mathcal{L}(m_t) \;=\; H,$$

so that $\boldsymbol{C}_t$ implements a Hessian-inverse preconditioner. Consequently, in our proof we focus solely on the mean update (7) (with $\boldsymbol{C}_t^{-1} \propto H$) and omit carrying the detailed covariance dynamics (8) through the convergence bounds.

Based on eq. (3), we sample P_i based on both the current mean value of the distribution parameter $\boldsymbol{\theta}_t$ and the sharpness aware term $\boldsymbol{\epsilon}^*$ obtained by optimizing the maximize problem $\max_{\|\epsilon\|_2 \le \rho} \mathcal{L}(P+\epsilon)$ within the nearly region around the current parameter P. Thus, by simplifying the second and higher order term with the variance of z, we can obtain the gradient of expectation for the loss $\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_t + \boldsymbol{\epsilon}^* + z)]$ as:

$$\nabla_{\boldsymbol{\theta}_t} \mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_t + \boldsymbol{\epsilon}^* + z)] = \nabla_{\boldsymbol{\theta}_t} \mathcal{L}(\boldsymbol{\theta}_t + \boldsymbol{\epsilon}^*) + \mathcal{O}(\delta_t^2)$$

Putting the above equation into eq. (7), we can obtain the natural gradient updating equation for the stage 1 of our SharpZO method, gives:

$$\begin{aligned}\theta_{t+1} &= \boldsymbol{\theta}_t \;-\; \sigma\, \mathbf{F}_{\boldsymbol{\theta}_t}^{-1}\, \nabla_{\boldsymbol{\theta}_t} \mathbb{E}\big[\mathcal{L}(P) \,\big|\, \boldsymbol{\theta}_t,\, \boldsymbol{C}_t\big] \\ &= \boldsymbol{\theta}_t - \sigma \mathbf{F}_{\boldsymbol{\theta}_t}^{-1} [\nabla \mathcal{L}(\boldsymbol{\theta}_t + \boldsymbol{\epsilon}^*) + \mathcal{O}(\delta_t^2)]\end{aligned}$$

Here, inspired by the proof of Therorem 4.1 of original SAM paper [3], we divide the natural gradient step of our sharpness-aware CMA-ES method into two steps:

$$\theta_{t+\frac{1}{2}} = \boldsymbol{\theta}_t + \rho \frac{\nabla \mathcal{L}(\boldsymbol{\theta}_t)}{\|\nabla \mathcal{L}(\boldsymbol{\theta}_t)\|_2}, \tag{9}$$

$$\theta_{t+1} = \boldsymbol{\theta}_t - \sigma \mathbf{F}_{\boldsymbol{\theta}_t}^{-1} [\nabla \mathcal{L}(\theta_{t+\frac{1}{2}}) + \mathcal{O}(\delta_t^2)] \tag{10}$$

Lemma 1 (Per–step error bound for sharpness-aware CMA-ES)**.** *Under Assumptions A2, the sharpness-aware CMA-ES method gives a per-step error bound for the updating process as:*

$$\mathcal{L}(\theta_{t+1}) \leq \mathcal{L}(\boldsymbol{\theta}_t) - \tfrac{1}{2}(1 - 2L\sigma^2)\|\nabla' \mathcal{L}(\boldsymbol{\theta}_t)\|^2 + L^2\rho^2 \;+\; L^3\sigma^2\,\rho^2$$

Proof. We begin our proof for the first stage defined in eq. (9). By the assumption of L-smoothness, we have:

$$\mathcal{L}(\theta_{t+\frac{1}{2}}) \leq \mathcal{L}(\boldsymbol{\theta}_t) + \big\langle \nabla \mathcal{L}(\boldsymbol{\theta}_t),\, \theta_{t+\frac{1}{2}} - \boldsymbol{\theta}_t \big\rangle + \tfrac{L}{2}\|\theta_{t+\frac{1}{2}} - \boldsymbol{\theta}_t\|^2.$$

Since $\theta_{t+\frac{1}{2}} - \boldsymbol{\theta}_t = \rho\, \nabla \mathcal{L}(\boldsymbol{\theta}_t)/\|\nabla \mathcal{L}(\boldsymbol{\theta}_t)\|$,

$$\mathcal{L}(\theta_{t+\frac{1}{2}}) \leq \mathcal{L}(\boldsymbol{\theta}_t) + \rho\, \|\nabla \mathcal{L}(\boldsymbol{\theta}_t)\| + \tfrac{L\rho^2}{2}. \tag{1}$$

For the second stage in eq. (9), we employ the similar idea and denote natural gradient as $\nabla' \mathcal{L}(\theta_t) \approx \mathbf{F}_{\boldsymbol{\theta}_t}^{-1}[\nabla \mathcal{L}(\theta_t)]$, which gives:

$$\begin{aligned}\mathcal{L}(\theta_{t+1}) &\leq \mathcal{L}(\boldsymbol{\theta}_t) - \sigma \left\langle \nabla' \mathcal{L}(\boldsymbol{\theta}_t),\, \nabla \mathcal{L}(\theta_{t+\frac{1}{2}}) \right\rangle + \tfrac{L\sigma^2}{2} \left\| \nabla \mathcal{L}(\theta_{t+\frac{1}{2}}) \right\|^2 \\ &\overset{(a)}{\leq} \mathcal{L}(\boldsymbol{\theta}_t) - \tfrac{1}{2}\|\nabla' \mathcal{L}(\boldsymbol{\theta}_t)\|^2 + \tfrac{1}{2}\left\|\nabla' \mathcal{L}(\boldsymbol{\theta}_t) - \nabla \mathcal{L}(\theta_{t+\frac{1}{2}})\right\|^2 + \tfrac{L\sigma^2}{2} \left\| \nabla \mathcal{L}(\theta_{t+\frac{1}{2}}) \right\|^2 \\ &\overset{(b)}{\leq} \mathcal{L}(\boldsymbol{\theta}_t) - \tfrac{1}{2}\|\nabla' \mathcal{L}(\boldsymbol{\theta}_t)\|^2 + L^2\rho^2 + \tfrac{L\sigma^2}{2} \left\| \nabla \mathcal{L}(\theta_{t+\frac{1}{2}}) \right\|^2 \\ &\leq \mathcal{L}(\boldsymbol{\theta}_t) - \tfrac{1}{2}\|\nabla' \mathcal{L}(\boldsymbol{\theta}_t)\|^2 + L^2\rho^2 + L\sigma^2\, \|\nabla' \mathcal{L}(\boldsymbol{\theta}_t)\|^2 \;+\; L^3\sigma^2\,\rho^2 \\ &\leq \mathcal{L}(\boldsymbol{\theta}_t) - \tfrac{1}{2}(1 - 2L\sigma^2)\|\nabla' \mathcal{L}(\boldsymbol{\theta}_t)\|^2 + L^2\rho^2 \;+\; L^3\sigma^2\,\rho^2\end{aligned}$$

where (a) is given by the fact $\langle a, b\rangle \geq \frac{1}{2}\|a\|^2 - \frac{1}{2}\|a - b\|^2$ with $a = \nabla \mathcal{L}(\boldsymbol{\theta}_t)$, $b = \nabla \mathcal{L}(\theta_{t+\frac{1}{2}})$ and (b) is given by eq. (9). □

D.2 Proof of Theorem 1

Now, we begin the proof of the global convergence rate of SharpZO method. Before we start, we first prove a per-step error bound for the stage 2 sparse ZO training in Lemma 2. Then, we perform inductive step based on the per-step error bound of the stage 2 and include the results in Lemma 1 as an initialization point of stage 2 training. Different from the proof in previous ZO fine-tuning paper [26] that consider an 'effective' rank for the dimension fo the optimization problem, we consider the true dimension d, as the trainable parameter in our prompt tuning case is much lower than the full model fine-tuning case. The proof of Lemma 2 is given as follows:

Lemma 2 (Per-step Error Bound for ZO-SGD)**.** *Given the Assumption A2 and the ZO-RGE gradient estimation follow eq. (4), by setting the learning rate $\eta \leq \frac{1}{2L(d+4)}$, we have:*

$$\mathbb{E}[\mathcal{L}(\boldsymbol{w}_{t+1}) \mid \boldsymbol{w}_t] \leq \mathcal{L}(\boldsymbol{w}_t) - \frac{\eta}{2}\|\nabla \mathcal{L}(\boldsymbol{w}_t)\|^2 + L\eta^2(\gamma^2(d+4) + \frac{L^2\mu^2}{4}(d+6)^3),$$

where d is the true parameter dimension of the trainable prompt $\boldsymbol{w}$ and L is the smoothness factor, μ is the ZO perturbation scal. The standard devation of the stochastic gradient estimation γ_t is defined as $\gamma_t = \mathbb{E}[\|\hat{\nabla} \mathcal{L}(\boldsymbol{w}_t) - \mathcal{L}_\mu(\boldsymbol{w}_t)\|^2]$, given the unbiased estimator $\hat{\nabla} \mathcal{L}(\boldsymbol{w}_t)$ for the smoothed objective function $\mathcal{L}_\mu(\boldsymbol{w})$.

Proof. Let $\boldsymbol{w}_t$ be the parameter at iteration t and we consider ZO-SGD using a gaussian smoothing estimator defined in eq. (4). Based on properties of $\mathcal{L}_\mu$ Theorem 3.1 (c) of [], the variance of the estimator satisfies:

$$\mathbb{E}[\|\hat{\nabla}\mathcal{L}(\boldsymbol{w}_t)\|^2] \leq 2(d+4)[\|\nabla\mathcal{L}(\boldsymbol{w}_t)\|^2 + \gamma^2] + \frac{\mu^2}{2}L^2(n+6)^3,$$

where is smoothness factor L is assumed

Given the learning rate $\eta > 0$, then from the smoothness of $\mathcal{L}$, the standard descent lemma gives:

$$\begin{aligned}\mathbb{E}[\mathcal{L}(\boldsymbol{w}_{t+1}) \mid \boldsymbol{w}_t] &\leq \mathcal{L}(\boldsymbol{w}_t) - \eta\|\nabla\mathcal{L}(\boldsymbol{w}_t)\|^2 + \frac{L}{2}\eta^2\mathbb{E}[\|\hat{\nabla}\mathcal{L}(\boldsymbol{w}_t)\|^2] \\ &\leq \mathcal{L}(\boldsymbol{w}_t) - \eta\|\nabla\mathcal{L}(\boldsymbol{w}_t)\|^2 + \frac{L}{2}\eta^2(2(d+4)[\|\nabla\mathcal{L}(\boldsymbol{w}_t)\|^2 + \gamma^2] + \frac{\mu^2}{2}L^2(d+6)^3)\end{aligned}$$

Rearranging:

$$\mathbb{E}[\mathcal{L}(\boldsymbol{w}_{t+1}) \mid \boldsymbol{w}_t] \leq \mathcal{L}(\boldsymbol{w}_t) - \eta\left(1 - L\eta(d+4)\right)\|\nabla\mathcal{L}(\boldsymbol{w}_t)\|^2 + L\eta^2(\gamma^2(d+4) + \frac{L^2\mu^2}{4}(d+6)^3).$$

Choose $\eta \leq \frac{1}{2L(d+4)}$ so that $1 - L\eta(d+4) \geq \frac{1}{2}$. Then:

$$\mathbb{E}[\mathcal{L}(\boldsymbol{w}_{t+1}) \mid \boldsymbol{w}_t] \leq \mathcal{L}(\boldsymbol{w}_t) - \frac{\eta}{2}\|\nabla\mathcal{L}(\boldsymbol{w}_t)\|^2 + L\eta^2(\gamma^2(d+4) + \frac{L^2\mu^2}{4}(d+6)^3).$$

□

Next, we proceed to prove Theorem 1 by performing an inductive argument based on the result of Lemma 2. Let the total number of steps in Stage 2 be denoted as $T_2 := T - T_c$. To facilitate the analysis, we define two suboptimality gap measures:

- $\Delta_t^{(1)} := \mathcal{L}(\boldsymbol{\theta}_t) - \mathcal{L}^*$, which denotes the optimality gap of the *distributional mean* $\boldsymbol{\theta}_t$ used in Stage 1 (Sharpness-aware CMA-ES);
- $\Delta_t^{(2)} := \mathcal{L}(\boldsymbol{w}_t) - \mathcal{L}^*$ used in Stage 2 (ZO optimization).

At the transition point $t = T_c$, Lemma 1 guarantees that the distributional mean θ_{T_c} satisfies a convergence bound on $\Delta_{T_c}^{(1)}$. Using a second-order Taylor expansion of the loss function around θ_{T_c}, we can relate the Stage 2 initialization gap $\Delta_{T_c}^{(2)}$ to $\Delta_{T_c}^{(1)}$ via:

$$\Delta_{T_c}^{(2)} = \mathcal{L}(p_{T_c}) - \mathcal{L}^* \leq \Delta_{T_c}^{(1)} + \rho\|\nabla\mathcal{L}(\theta_{T_c})\| + \frac{L}{2}\left(\rho^2 + \delta_{T_c}^2\,\mathrm{Tr}(C_{T_c})\right). \tag{11}$$

This inequality provides the initial condition for the inductive proof in Stage 2, where we now track the evolution of $\Delta_t^{(2)}$ for $t = T_c, \ldots, T$, as governed by Lemma 2. Here, we first bound $\Delta_{T_c}^{(1)}$

By Lemma 1, for each t we have

$$\mathcal{L}(\theta_{t+1}) \;\leq\; \mathcal{L}(\boldsymbol{\theta}_t) \;-\; \tfrac{1}{2}\big(1 - 2L\sigma^2\big)\,\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\|^2 + L^2\rho^2 + L^3\sigma^2\rho^2.$$

Subtracting $\mathcal{L}^*$ from both sides yields

$$\Delta_{t+1} \;\leq\; \Delta_t \;-\; \tfrac{1}{2}\big(1 - 2L\sigma^2\big)\,\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\|^2 + C, \quad C := L^2\rho^2 + L^3\sigma^2\rho^2.$$

Under the PL inequality $\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\|^2 \geq 2\mu\,\Delta_t$, it follows that

$$\begin{aligned}\Delta_{t+1}^{(1)} &\leq \Delta_t^{(1)} - \tfrac{1}{2}\big(1 - 2L\sigma^2\big)\,(2\mu\,\Delta_t) + C \\ &= \big[1 - \mu(1 - 2L\sigma^2)\big]\,\Delta_t + C.\end{aligned}$$

Set

$$\xi \;:=\; \mu\,(1 - 2L\sigma^2), \quad 0 < \xi < 1 \quad \text{(assuming } \sigma^2 < 1/(2L)\text{)}.$$

Then the recursion becomes

$$\Delta_{t+1}^{(1)} \;\leq\; (1 - \xi)\,\Delta_t^{(1)} + C.$$

Unrolling this for $t = 0, 1, \ldots, T_c - 1$ gives

$$\Delta_{T_c}^{(1)} \;\le\; (1-\xi)^{T_c}\,\Delta_0^{(1)} + C\sum_{i=0}^{T_c-1}(1-\xi)^i \;=\; (1-\xi)^{T_c}\,\Delta_0^{(1)} + \frac{C}{\xi}\Big[1-(1-\xi)^{T_c}\Big].$$

Substituting back $C = L^2\rho^2 + L^3\sigma^2\rho^2$ and $\xi = \mu(1-2L\sigma^2)$ completes the proof:

$$\Delta_{T_c}^{(1)} \;\le\; (1-\xi)^{T_c}\,\Delta_0^{(1)} \;+\; \frac{L^2\rho^2 + L^3\sigma^2\rho^2}{\mu\,(1-2L\sigma^2)}\Big[1-(1-\xi)^{T_c}\Big]. \tag{12}$$

Now, we begin to prove the global convergence rate for the SharpZO method. Let's focus back into the bound given in Lemma 2. By the PL-inequality assumed in Assumption A1, we have:

$$\begin{aligned}\mathbb{E}\big[\mathcal{L}(\boldsymbol{w}_{t+1}) \mid \boldsymbol{w}_t\big] &\le \mathcal{L}(\boldsymbol{w}_t) - \frac{\eta}{2}\,\|\nabla\mathcal{L}(\boldsymbol{w}_t)\|^2 + C_{\mathrm{var}}(d,L)\,\eta^2 \\ &\le \mathcal{L}(\boldsymbol{w}_t) - \eta\mu\,\big(\mathcal{L}(\boldsymbol{w}_t) - \mathcal{L}^*\big) + C_{\mathrm{var}}(d,L)\,\eta^2,\end{aligned}$$

where $C_{\mathrm{var}}(d,L) = \frac{L(d+4)+\frac{L^2\mu^2}{4}(d+6)^3}{\eta}$ is some constant. Taking full expectation and with $\Delta_t^{(2)} := \mathbb{E}[\mathcal{L}(\boldsymbol{w}_t)] - \mathcal{L}^*$ gives the one-step contraction

$$\Delta_{t+1}^{(2)} \;\le\; (1-\eta\mu)\,\Delta_t^{(2)} \;+\; C_{\mathrm{var}}(d,L)\,\eta^2.$$

Given the current step t, we define $t_2 := t - T_c$ and unroll this linear recursion for $t = T_c, \ldots, T_c + t_2$ yields

$$\begin{aligned}\Delta_t^{(2)} &\le (1-\eta\mu)^{t_2}\,\Delta_{T_c}^{(2)} + \frac{C_{\mathrm{var}}(d,L)\,\eta}{\mu}\Big(1-(1-\eta\mu)^{t_2}\Big) \\ &\overset{(a)}{\le} (1-\eta\mu)^{t_2}(\Delta_{T_c}^{(1)} + \rho\|\nabla\mathcal{L}(\theta_{T_c})\| + \frac{L}{2}\left(\rho^2 + \delta_{T_c}^2\,\mathrm{Tr}(C_{T_c})\right)) + \frac{C_{\mathrm{var}}(d,L)\,\eta}{\mu}\Big(1-(1-\eta\mu)^{t_2}\Big) \\ &\overset{(b)}{\le} (1-\eta\mu)^{t_2}((1-\xi)^{T_c}\,\Delta_0^{(1)} \;+\; \frac{L^2\rho^2 + L^3\eta^2\rho^2}{\mu\,(1-2L\eta^2)}\Big[1-(1-\xi)^{T_c}\Big] \\ &\quad + \rho\|\nabla\mathcal{L}(\theta_{T_c})\| + \frac{L}{2}\left(\rho^2 + \delta_{T_c}^2\,\mathrm{Tr}(C_{T_c})\right)) + \frac{C_{\mathrm{var}}(d,L)\,\eta}{\mu}\Big(1-(1-\eta\mu)^{t_2}\Big),\end{aligned}$$

where (a) is given by eq. (11) and (b) follows the bound of $\Delta_{T_c}^{(1)}$ proved in eq. (12). Finally, by ensuring $\Delta_t^{(2)} \le \epsilon$, we have:

$$t_2 = \mathcal{O}(\frac{1}{\eta\mu}\ln\frac{X-B}{\epsilon-B}),$$

where

$$X = (1-\xi)^{T_c}\,\Delta_0^{(1)} + \frac{L^2\rho^2 + L^3\eta^2\rho^2}{\xi}\Big[1-(1-\xi)^{T_c}\Big] + \rho\|\nabla\mathcal{L}(\theta_{T_c})\| + \frac{L}{2}\big(\rho^2 + \delta_{T_c}^2\,\mathrm{Tr}\,C_{T_c}\big) \tag{13}$$

$$= (1-\xi)^{T_c}\,\Delta_0^{(1)} + \rho^2((L^2 + L^3\eta^2)(\xi-2) + \frac{1}{2}L) + \rho\|\nabla\mathcal{L}(\theta_{T_c})\| + \frac{L\delta_{T_c}^2\,\mathrm{Tr}\,C_{T_c}}{2} \tag{14}$$

and

$$B = \frac{C_{\mathrm{var}}(d,L)\,\eta}{\mu}, \quad C_{\mathrm{var}}(d,L) = \frac{L(d+4)+\frac{L^2\mu^2}{4}(d+6)^3}{\eta} \tag{15}$$

Here, if we focus on the influence of smoothness factor L and ignoring the lower-order terms for convenience, we can write the convergence rate t_2 as:

$$t_2(L) \approx \mathcal{O}\left(\frac{1}{\eta\mu}\log\left(\frac{L^3\eta^2\rho^2}{\epsilon}\right)\right) \tag{16}$$

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed the limitation in Appendix A.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.

- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We prodive detailed description for the assumption and proof.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We included the detiled hyper-parameter setup in the appendix

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully

might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We will release the code, the anonymous code is provided in the supplemental material with documentation

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

Justification: We use public dataset and we have offered detailed instruction on the hyperparameter of the data processing.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: In the training curve we provided, we mark the variance interval by shallow

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide wall data and memory information in the description.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

Justification: We have checked.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We have cited the related papers.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We provided such details

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.